CLEARINGHOUSE FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION CFSTI
DOCUMENT MANAGEMENT BRANCH 410.11

LIMITATIONS IN REPRODUCTION QUALITY

ACCESSION # 604 299

604 299

☒ 1. WE REGRET THAT LEGIBILITY OF THIS DOCUMENT IS IN PART UNSATISFACTORY. REPRODUCTION HAS BEEN MADE FROM BEST AVAILABLE COPY.

□ 2. A PORTION OF THE ORIGINAL DOCUMENT CONTAINS FINE DETAIL WHICH MAY MAKE READING OF PHOTOCOPY DIFFICULT.

□ 3. THE ORIGINAL DOCUMENT CONTAINS COLOR, BUT DISTRIBUTION COPIES ARE AVAILABLE IN BLACK-AND-WHITE REPRODUCTION ONLY.

□ 4. THE INITIAL DISTRIBUTION COPIES CONTAIN COLOR WHICH WILL BE SHOWN IN BLACK-AND-WHITE WHEN IT IS NECESSARY TO REPRINT.

□ 5. LIMITED SUPPLY ON HAND: WHEN EXHAUSTED, DOCUMENT WILL BE AVAILABLE IN MICROFICHE ONLY.

□ 6. LIMITED SUPPLY ON HAND: WHEN EXHAUSTED DOCUMENT WILL NOT BE AVAILABLE.

□ 7. DOCUMENT IS AVAILABLE IN MICROFICHE ONLY.

□ 8. DOCUMENT AVAILABLE ON LOAN FROM CFSTI (TT DOCUMENTS ONLY).

□ 9.

PROCESSOR: DM

1

COMPUTATIONAL EXPERIENCE IN SOLVING LINEAR PROGRAMMING PROBLEMS

Wm. Orchard-Hays

P-482 ✓ B

15 March 1954

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA

SUMMARY

Linear programming techniques in solving [illegible]

Many administrative and planning problems — as well as others — can be studied and solved with mathematical models by the technique known as linear programming. The most practical way yet devised of solving the resulting mathematical problem is the so-called simplex method.

It is to be emphasized that comparatively simple models quickly lead to large-scale computations. Indeed, computing machinery adequate for such work has only existed in recent years. Current work is being done on an IBM 701 calculator, based on experience gained with smaller models solved on the IBM CPC. One of these was a fairly thorough study of a petroleum blending problem which illustrates the power of linear programming as a tool for planning. Many technical problems yet remain to be resolved but machine codes now being perfected will vastly increase the versatility of the present methods. Those who would use the technique for their own problems, however, should give thought to gaining a more precise notion of the relative values of the activities and 'goods' to be studied.

COMPUTATIONAL EXPERIENCE IN SOLVING LINEAR PROGRAMMING PROBLEMS

THE LINEAR PROGRAMMING TECHNIQUE

One of the mathematical tools that has been applied to Operations Research, or at least to some areas of it, is the technique known as linear programming. This may be defined briefly as follows: the maximization of a linear form subject to linear inequality restraints. The word 'maximization' can of course be replaced with 'minimization' and a better, more general term is 'optimization'. Although the foregoing definition is rigorous enough, it is somewhat barren as far as conveying to the mind the scope of problems to which the technique may be applied. It has been remarked that the term 'linear' is unfortunate in that it sounds unduly restrictive. Actually, whether or not a problem is linear depends to some extent on how it is formulated. As one becomes more familiar with the method, the apparent range of problems to which it can be applied is enlarged. We will elaborate on this later.

The purpose in constructing a linear programming model is of course analogous to the purpose in making miniature, but reasonably true-to-scale, models in planning the lay-out of a home, office, or shop. In the latter case, there is so much space of a certain geometric shape available, there are so many pieces to be put in it, and there are certain requirements as to utility, access room, and so forth, which must be satisfied. By moving the models about, we can arrive at the best arrangement even before the real machines arrive.

In a similar way we may construct a linear programming model of the activities involved in some enterprise. There is difficulty in manipulating a model of this kind, however, due to the fact that its geometric interpretation is not three-dimensional. We are concerned with many activities, each having several aspects and inter-relationships with the others, so that intuition soon bogs down. We simply cannot manipulate the various rows and columns of the model matrix, by hand so to speak, and keep track of the consequences. Instead, formal mathematical procedures must be developed and followed, both to insure that no requirements have been violated and to make the computations feasible.

THE SIMPLEX METHOD

It is almost universally agreed that the best procedure yet devised — and in my own opinion likely to be devised — for solving linear programming problems, as such, is the simplex method. This was originally developed by Dr. George B. Dantzig who is now at RAND and with whom I have worked for a year and a half in developing machine methods for the simplex process. Since this has been by far the largest part of my experience with linear programming, I may use the two terms as though they were synonymous, in spite of the fact that there are other ways to solve these problems.

Perhaps the most instructive way to explain the technique in a few minutes is to construct a model for a very simple, transparent, and necessarily extremely over-simplified problem.

CONSTRUCTING A MODEL

Suppose you had six working days in which to write a report. Let us assume that 1½ days to assemble data and 2 days for writing are minimum requirements. Besides this, time to evaluate the data and devise illustrations is desirable. How should you schedule your time to produce the best results?

Four activities were enumerated above:

A: Assemble data; minimum of 3/2 days.
E: Evaluate data.
D: Devise illustrations and examples.
W: Write report; minimum of 2 days.

Each activity will be represented by a column of figures, that is a vector, and by using another column for the right hand side, we have 3 inequations in time:

$$\begin{array}{lccccl}
\text{Activity} & A & E & D & W & \\
\text{Total time spent} & 1\cdot t_A + & 1\cdot t_E + & 1\cdot t_D + & 1\cdot t_W & \leq 6 \\
\text{Time to assemble data} & 1\cdot t_A & & & & \geq 3/2 \\
\text{Time to write report} & & & & 1\cdot t_W & \geq 2
\end{array}$$

the t's being the variables in the problem. Now it is inconvenient to work with inequalities so we will introduce three additional activities:

S: Total time not used, a slack vector.
B: Time spent assembling data beyond minimum required.
C: Time spent writing beyond minimum required.

Then, if we omit the t's and plus signs as being understood, we have:

$$(1)\quad \begin{array}{cccccccl}
A & E & D & W & S & B & C & \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & = 6 \\
1 & 0 & 0 & 0 & 0 & -1 & 0 & = 3/2 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & = 2
\end{array}$$

Since we always require activities to be used in non-negative amounts, clearly B and C can have non-zero coefficients only after the two requirements are met.

However, (1) only represents the inequality restraints which are imposed on the system; we still lack the most important linear form, the one which maximizes the worth of the finished report. To specify this form we will have to decide on the relative worth of the different activities. For the sake of argument, let us suppose that every hour spent evaluating the data will contribute most to the final result, that every hour spent on illustrations and examples will contribute about half as much worth as the hour spent evaluating, and that each additional hour spent assembling data or writing will each contribute about a fourth as much worth as the evaluating. The slack activity is worth nothing and since specific demands are made on A and W, we need only assign values to E, D, B, and C. If v_i is the relative value of the i-th activity, then we wish to have

$$t_E v_E + t_D v_D + t_B v_B + t_C v_C = \text{maximum}$$

or incorporating this in our system as the top row (usually indexed zero):

$$\begin{array}{lccccccccl}
 & A & E & D & W & S & B & C & & \\
 & 0 & 1 & 1/2 & 0 & 0 & 1/4 & 1/4 & = & \text{max} \\
 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & = & 6 \\
(2) & 1 & 0 & 0 & 0 & 0 & -1 & 0 & = & 3/2 \\
 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & = & 2
\end{array}$$

(Another possible constraint)

$$\begin{array}{lccccccccl}
(2a) & 1 & -1 & 0 & 0 & 0 & 1 & 0 & = & 0
\end{array}$$

This last system is a perfectly good linear programming model which the simplex method will solve in short order. Of course the obvious solution is to spend 1½ days assembling data, 2½ days evaluating it, and 2 days writing it up. The total relative value for this schedule is greater than that for any other schedule. It is in general true that any system of m equations (not including the form to be optimized) has an optimal solution involving not more than m activities provided it has any optimal solution at all. If we wrote another equation or two specifying more subtle inter-relationships between the activities, then the solution would not be quite so obvious. For example, we might require that $t_A + t_B = t_E$. (See (2a)).

COMPUTATIONAL EXPERIENCE ON THE CPC (IBM's Card Programmed Calculator)

It must be emphasized that comparatively simple models quickly lead to large-scale computations. I think it is fair to say that the application of the simplex method, or any other linear programming method, to real-life problems is practical only if one has a large, high-speed computer available, except in certain special cases. A great deal of effort is currently being expended on developing adequate machine codes for quite extensive problems. This interest is not confined to one group or one area, but companies of various kinds all over the country are devoting time and money to learning how to use either linear programming or some comparable technique, such as combinatorial analysis. A little later I shall indicate what sort of codes we are developing at RAND for the IBM 701 calculator. Certain problems, however, have been studied quite extensively on a now-so-humble machine as the CPC. I would like to discuss one such problem at some length.

To begin with, I should indicate what size problem we proposed to handle on the CPC. It has become practically standard practice to denote the number of rows — or equations —

of a model matrix by m, and the number of columns — or activities — by n. Although both m and n, as well as some unknown or indefinite factors, enter into our formula for the time required to solve a simplex problem, it is the size of m which is critical for space as well as time. Along about December of 1952, I began developing a set of plug-boards and a procedure for the CPC to handle systems up to m = 27 and n = 70. This was my first exposure both to linear programming and the simplex method. We ran a problem about the first of May, 1953. The development of a CPC set-up had been intended as a training step and for this purpose the time was well spent; however, for practical purposes, the set-up was a failure. Its first drawback was that it was much too slow, even for the CPC. I think I can fairly claim that this was not due to a lack of ingenuity on my part in using the machine, but rather was due to the unwieldy amount of information we were attempting to maintain with the very limited internal storage available. The realization of this fact led to a major revision in the computational procedure which has proved very fruitful, and in fact we have not yet discovered all the possibilities which it affords. Mathematically, this revision consists of replacing the inverse of an m x m matrix by the product of elementary column matrices. Those who are interested in this technique will find references listed at the end of the paper.

The second difficulty with this first set-up was its inflexibility. I attempted to make the operation as automatic as possible and it was some time before I realized that this was a stumbling-block. I remark on this to re-emphasize the need for high-speed, internally-programmed computers. To attain any degree of automation on a machine like the CPC, one must adapt his procedures to the idiosyncrasies of the machine. I do not mean to discredit the CPC; it has, and is, performing much valuable work. But it is not suited to a mixture of data handling, computation, and logical juggling — such as the simplex method is — unless one is willing to stand at the card hopper and make complicated card manipulations as the problem progresses, and to make continual visual checks, an inefficient procedure at best.

A second CPC set-up was devised capable of handling systems up to $m = 40$ and $n = 99$. As much card-handling and as many decisions as seemed reasonably safe were imposed on the machine operator. The set-up is still in use for certain specialized problems and the procedure has become the logical frame-work for our 701 codes. Several problems, one as large as $m = 36$ and $n = 99$, were run with excellent results. However, the time required for a problem as large as just mentioned is likely to be a few days. Probably 50% of the work done by this set-up was on a 26 x 61 matrix which was later expanded to 28 x 68 which seems to be about optimal size for the CPC, the machine time required being of the order of a working day. This problem, or really series of problems, constituted the computations needed in an economic study made by A. S. Manne concerning the output of petroleum products in a thermal cracking refinery. The results of this study are included in a book being published which is listed in the references. I am indebted to Dr. Manne for the accompanying chart and the following information concerning the construction of the model and the interpretation of the results obtained.

A PETROLEUM BLENDING PROBLEM

Fig. 1 is a schematic drawing of the refinery, a hypothetical plant in the Oklahoma region. The input consists of 38° API gravity Mid-Continent crude and there are ten end products as shown. At the left is the primary distillation column which takes the crude and produces gases, straight-run gasoline, three weights of distillate oil, and residue. The distillates and residue may all be used directly in fuel oil blending and in addition each feeds a cracking coil, the four coils in turn feeding a secondary fractionating column. The secondary column produces the same cuts as the primary column and the cracked distillates may be recycled to the coils or used in fuel oil blending. The cracked gasoline goes to the gasoline blending unit as did the straight-run gasoline, and here tetra-ethyl lead (TEL) is also added.

Time will not permit further explanation of the technology involved; the details are available in the reference previously mentioned for those who have a technical interest in the petroleum industry. Suffice it to say that the equations required to specify all the conditions suggested by the diagram were not all linear, initially. In particular, there were seven independent variables — four representing the proportion of the straight-run streams used for cracking stock and three representing the recycle ratios of the secondary distillate oils — which entered all equations in a non-linear manner. Using a suggestion of

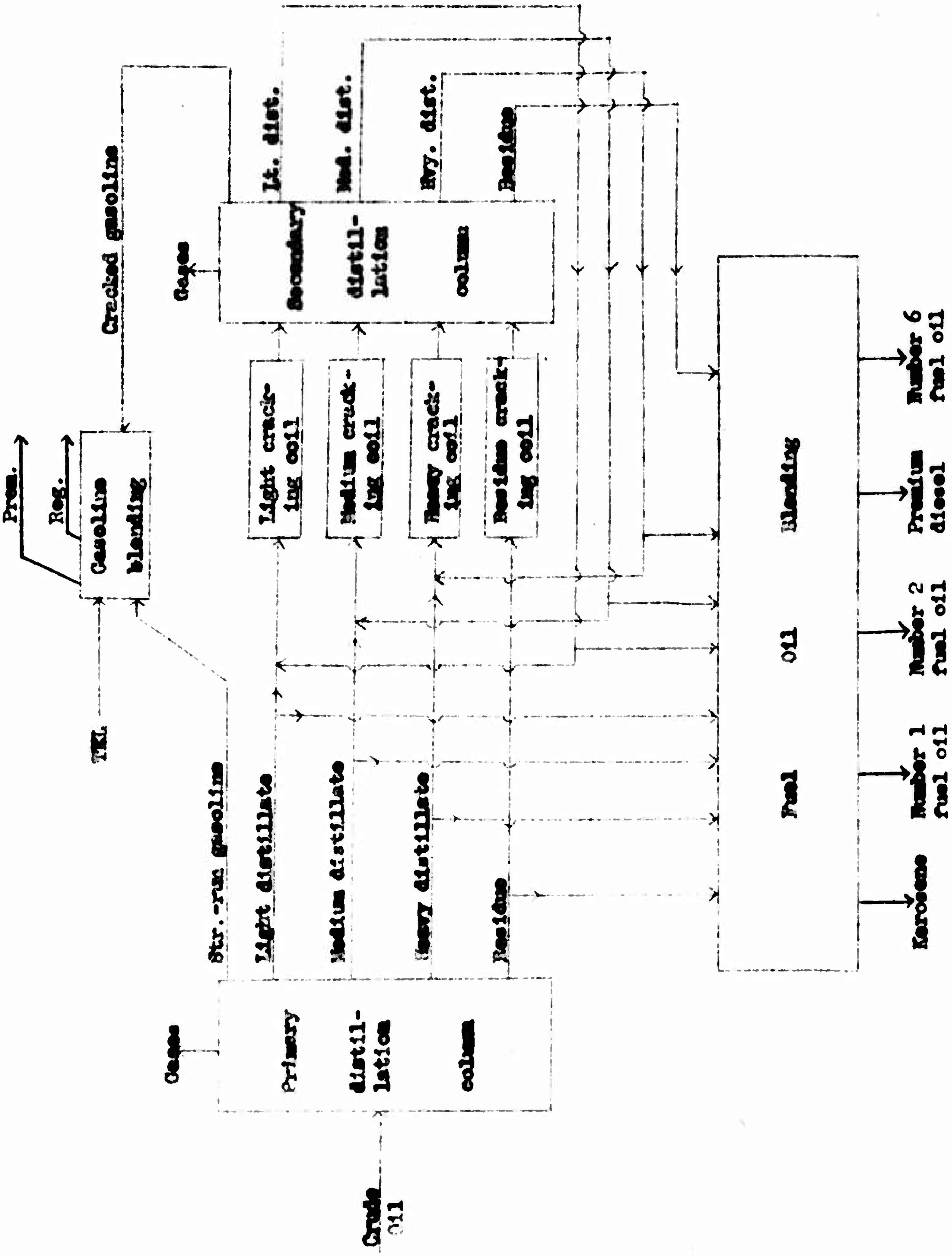
Crude oil
Primary distillation column
Gases
Str.-run gasoline
Light distillate
Medium distillate
Heavy distillate
Residue
TEL
Gasoline blending
Prem.
Reg.
Light cracking coil
Medium cracking coil
Heavy cracking coil
Residue cracking coil
Secondary distillation column
Gases
Cracked gasoline
Lt. dist.
Med. dist.
Hvy. dist.
Residue
Fuel Oil Blending
Kerosene
Number 1 fuel oil
Number 2 fuel oil
Premium diesel
Number 6 fuel oil

Fig. 1

Kenneth Arrow, Manne defined seven new independent variables which do enter in a linear manner. It seemed worthwhile to make special mention of this because it illustrates how established technical usage is sometimes a stumbling-block in the way of simplification of a problem. The original variables represented recycle ratios, which I understand is the concept generally employed by petroleum engineers. The new variables represent quantities, but the change in definition does not alter any technical assumptions while it does render the calculations considerably easier to perform.

The only non-linearities remaining were those concerning the blending of TEL in gasoline. Since the allowable variation in TEL for premium grade gasoline is very narrow, this was fixed at a constant value. The TEL level in regular gasoline was handled as a parameter and three different optimal schedules were obtained for the three different values. As Manne later discovered, he could have defined alternative activities incorporating the recipes for different levels of TEL in both regular and premium gasoline instead of using an activity of blending. This would have eliminated the need for separate optimizations, and additional concentrations for premium gasoline could have been incorporated. Besides this, one equation out of the 26 required to define all conditions could have been eliminated. However, as it turned out, the TEL level was not a very important consideration anyway.

The model was beautifully set up with 61 activities involving the 26 equations. In addition, there was a form to be minimized which was changed for different stages of the study, or as Manne calls them, the four 'runs'. From my viewpoint — that is that of a numerical analyst — each of his 'runs' constituted several runs, or problems, but in the future we hope to have more elegant methods which will not seem so piecemeal. In fact, this one study, or series of related simplex problems, led to the development of numerical techniques which greatly expand the usefulness of the simplex method. It is now possible — and indeed an accomplished fact — to build back into the simplex codes for large machines, such as the 701, the automation which was deliberately sacrificed for the second CPC set-up, plus a great deal more whose equivalent on the CPC, regardless of time, was laborious or impractical.

The first set of calculations assumed an existing plant and an existing market price structure. The results suggested that the gasoline blending problem could be divorced entirely from the rest of the refinery operations and that the highest concentration of TEL allowable was the most profitable. Consequently, in succeeding calculations, this parameter was left fixed at its highest value. The general results of these first computations checked very well with actual conditions obtaining in the Oklahoma-Kansas-Missouri region at the time on which the assumptions were based.

Forty-three iterations involving about ten machine hours were required to obtain the first optimal solution, for the lowest TEL level. We found it possible to obtain solutions for the other levels by continuing the calculations a very few iterations — using appropriate tricks — rather than by starting over from scratch. This was our first attempt — somewhat crude — at what Manne calls parametric linear programming and what I call post-optimality problems — reflecting, I suppose, the different viewpoints of one who formulates a problem and one who carries through the computations. Recently, in checking out a 701 code incorporating double-precision, floating-point arithmetic and designed to handle problems up to m = 100 or even higher, I re-solved the problem for the highest TEL level directly. This required 41 iterations and about 22 minutes. It was gratifying to note that the results checked with the round-about CPC run to better than four decimal places for the variables and to about three for the shadow prices, which are the prices imputed to the commodities by the system of activities engaged in.

These shadow prices are very interesting and useful in their own right — nearly as much so as the values of the variables. They are a free by-product of the simplex method since they constitute the elements of a certain row of the inverse of the optimal basis matrix — that is, the square matrix obtained by adjoining the columns of activities used in the optimal schedule. This row from the inverse of any basis is used in the criterion for determining whether any other basis gives an improvement. Consequently, this row vector from the optimal basis can be used for pricing out any other activity not originally included in the problem. The elements are the prices imputed to the various items represented by the rows of the model when the schedule involves the activities making up the basis, and these prices are independent of the levels at which the activities are used.

The second set of calculations was to study the effect of a market rise in the price of Number 6 fuel oil on an optimal program. Here we developed an important type of parametric programming involving changes in the price structure, that is varying the values placed on different activities. It turned out that if the price of Number 6 fuel oil was quadrupled, the increase in production should only be 10%, although there were some side effects on other distillate oil blends.

The third computation was similar to the second but used variations in the values of gasoline and Number 1 fuel oil as reflections of changes in demand of these two important items in order to study production flexibility within a given plant. The results showed that, regardless of demand, the plant could not convert more than about 22% of its crude into either product. This technique of varying a price structure in order to exert strong pressures without actually imposing rigid demands is an important aspect of linear programming. Not the least of its uses is in determining what values <u>should</u> be placed on activities to simulate real-life conditions, where the value of an activity is not such a simple thing as the going market price of an item. That is, linear programming can be used in somewhat reverse fashion with known results to impute values.

For the fourth set of computations, two more equations and seven more activities were added. It was later discovered that five equations and six activities from the total could have been eliminated. Whatever loss in machine time this may have incurred, however, was surely more than compensated for by the experience gained in applying linear programming. Dr. Manne has recently developed a 100 order model to be solved on the 701. Undoubtedly the experience he gained with the smaller model used on the CPC has been invaluable in formulating his new model. I am sure that without the experience I gained in solving problems on the CPC, it would have been impossible for me to develop the 701 code which will solve the 100 row system. I strongly recommend that any company wishing to use linear programming extensively set up informal teams embracing economics, business administration or some appropriate subject together with numerical analysis and machine know-how, and have them begin with solving small order problems to gain familiarity with the technique and to de-bug their procedures and methods of communication.

The fourth run was in some ways the most interesting of all. It was a capital investment problem involving the construction of an all-new refinery. The results showed that, for the same original investment of $1.8 millions, the refinery could have been earning $132,900 more per annum; or, conversely, to realize the same profit, the initial investment could have been trimmed to $1.45 millions. These results admittedly were dependent on estimated construction costs which might vary widely. However, the shadow price of invested capital showed that small increases in the capital outlay would have yielded returns of about 21% per annum.

From the standpoint of numerical analysis, run four was interesting in that it made use of a parameter — perhaps better termed a variable — in the right hand side. This variable was held fixed until an optimal solution was obtained. Then the critical values of the variable were computed and corresponding changes made in the optimal schedule as these critical values were crossed. We were able to develop a fixed and straight-forward procedure for these calculations. This consists of using the desired change as a pseudo-activity to determine a certain index and then using the row of the inverse matrix denoted by this index for pricing out. Single changes in any number of the right hand elements may be considered simultaneously. There are other applications of this device which have not yet been thoroughly explored.

PRESENT AND PLANNED SIMPLEX CODES FOR THE 701

I began last summer to develop a code for the 701 to carry out the simplex code for larger models, but without any parametric variations. George Dantzig and I felt that the system used on the CPC should provide an extremely fast code for the 701. Unfortunately, I went overboard on the idea of speed, and neglected the matter of accuracy. Working intermittently on the project, I did not get it checked out until the end of January, this year. It was a whiz-bang for speed, sure enough. It plowed through a 25 order test problem in about a minute. The only trouble was that when the numbers were anything but low order integers, round-off error accumulated so fast that the code would bog down in a few iterations. This so distressed me that I worked practically day and night throughout February writing a new code incorporating double-precision, floating-point arithmetic. (After all, I had to have something to say at this point.) The esults were worth the effort however.

Although the speed is not remarkable, considering the machine, the code will handle larger problems more quickly than any other which has been reported to date. The accuracy is excellent. Several stringent tests are built into the code. One of these is that after each ten iterations, the solution is actually multiplied out as a matrix-vector product, and compared with the right hand side. After 100 iterations on an 88 by 174 model, the solution agreed exactly with the right hand side to seven decimals which is all that are printed. The 100 iterations on this fair-sized problem required 2-1/2 hours. I learned enough from the mistakes made in the first code to enable me to arrange the new one in such a manner that variations can be introduced quite easily. Several auxiliary codes to be used in various circumstances have already been checked out and it is planned to build up a complete system of coding to handle all procedures with which we are familiar to date. The code can be immediately changed to handle 200 order systems provided the accuracy maintained proves sufficient.

It is hoped that a practical procedure may soon be developed for handling dynamic models of larger magnitude. This is essential to the solution of large and complicated scheduling problems set up in linear programming form. The difficulties to be overcome, however, loom as tremendous at this time. Perhaps a year hence, they will not seem so formidable, but we will have a try at it in any event.

In conclusion, may I again emphasize that in synthesizing any complicated enterprise, we must have precise notions of the relative values of the components to be considered. As long as models represent strictly physical conditions, the cost row is likely to be merely another set of restraints handled in a special way. However, when a model involves quantities dependent on human relations, the values to be imputed may be a highly complex matter. I suggest that this is the area requiring the most study by those in the field of management.

REFERENCES

Koopmans, T. (ed.), ACTIVITY ANALYSIS OF PRODUCTION AND ALLOCATION, New York, John Wiley and Sons. (Basic reference to linear programming, the associated geometry, and general mathematical background.)

Charnes, Cooper, and Henderson, AN INTRODUCTION TO LINEAR PROGRAMMING, New York, John Wiley and Sons. (Economic introduction and mathematical theory including a presentation of the simplex method in its original form.)

Notes on Linear Programming, The RAND Corp. Papers, Santa Monica, Calif. Part I: THE GENERALIZED SIMPLEX METHOD FOR MINIMIZING A LINEAR FORM UNDER LINEAR INEQUALITY RESTRAINTS, P-392, by George B. Dantzig, Alex Orden, and Philip Wolfe.
Part II: DUALITY THEOREMS, P-393, by Dantzig and Orden.
Part III: COMPUTATIONAL ALGORITHM OF THE SIMPLEX METHOD, P-394, by Dantzig.
Part IV: AN ALGEBRAIC PROOF OF THE MINMAX THEOREM, P-409, by Dantzig.
Part V: ALTERNATE ALGORITHM FOR THE REVISED SIMPLEX METHOD USING A PRODUCT FORM FOR THE INVERSE, P-460, by George Dantzig and Wm. Orchard-Hays.
Part VI: THE RAND CODE FOR THE SIMPLEX METHOD (for the IBM 701), P-477 by Orchard-Hays.
Part VII: THE DUAL SIMPLEX ALGORITHM, P-472 by Dantzig.

Manne, A. S., PETROLEUM REFINERY OPERATIONS SCHEDULING, (a forthcoming book). Chapter VI, from which much of the material in the text was drawn, is available as a paper, P-493, The RAND Corp., Santa Monica, California.

________, NOTES ON PARAMETRIC LINEAR PROGRAMMING, P-468, The RAND Corp.

Lemke, C. E., THE DUAL METHOD OF SOLVING THE LINEAR PROGRAMMING PROBLEM, Carnegie Institute of Technology, Department of Mathematics, Technical Report No. 29.

A. Hoffman, M. Mannos, D. Sokolowsky, and N. Wiegmann, COMPUTATIONAL EXPERIENCE IN SOLVING LINEAR PROGRAMS, N.B.S. Report 2501, May 15, 1953.